

# Programmer's Learning Machine

Matthieu Nicolas

IJD Seminar, 2016-02-02

## 1 Presentation of PLM

- Purposes
- Demo
- About PLM
- Architecture

## 2 User's code's assessment

- Challenges
- Extraction of the execution component
- Isolation and virtualization

## 3 Result

## 4 Next steps

## 1 Presentation of PLM

- Purposes
- Demo
- About PLM
- Architecture

## 2 User's code's assessment

- Challenges
- Extraction of the execution component
- Isolation and virtualization

## 3 Result

## 4 Next steps

# Presentation of PLM

## Purposes

- Application to learn programming.

# Presentation of PLM

## Purposes

- Application to learn programming.
- Allows students to progress at their own speed...

# Presentation of PLM

## Purposes

- Application to learn programming.
- Allows students to progress at their own speed...
- ... while the teacher helps the ones having trouble.

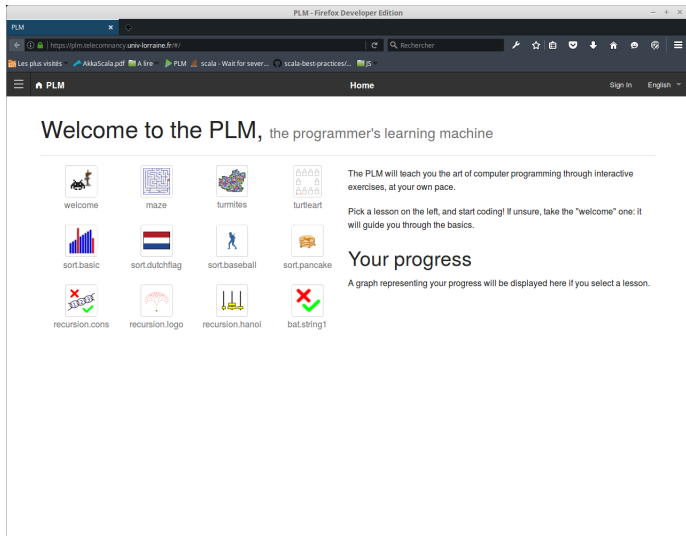
# Presentation of PLM

## Purposes

- Application to learn programming.
- Allows students to progress at their own speed...
- ... while the teacher helps the ones having trouble.
- Used at TELECOM Nancy since 2008.

# Presentation of PLM

## Quick demo





# Presentation of PLM

12 lessons, 200 exercises

PLM - Mozilla Firefox

https://plm.univ-lorraine.fr/la/lessons/maze/maze-wallfollower/maze-wallfollower-maze

Les plus visités Linux Mint Community Forums Blog News NinjaMock - free tool... Spotify Web Player Wau The Most Amaz... Coursa P2P

PLM Maze / WallFollowerMaze Se connecter Java Français

World Objective



Résultats de l'exécution

Éditeur de code:

```
public void run() {  
    // ...  
}
```

Cas de tests:

Another labyrinth Labyrinth

# Presentation of PLM

12 lessons, 200 exercises

PLM - Mozilla Firefox

https://plm.univ-lorraine.fr/.../recursion/hanoi/hanoi.lessons/hanoi.Hani

Rechercher

Les plus visités Linux Mint Community Forums Blog News NinjaMock - free tool... Spotify Web Player Wau The Most Amaz... Coursa P2P

PLM Recursion.hanoi / HanoiBoard Se connecter Java Français

World Objective

0 Move



Reinitialiser API

Éditeur de code:

```
public void hanoi(int height, int src, int ether, int dst) {  
    // ...  
}
```

Résultats de l'exécution

Cas de tests:

solve(0,1,2) solve(1,2,0) solve(2,0,1)

# Presentation of PLM

12 lessons, 200 exercises

PLM - Mozilla Firefox

https://plm.univ-lorraine.fr/ta/lessons/recursion/logo/recursion/logo/tree.Tre

Les plus visités Linux Mint Community Forums Blog News NinjaMock - free tool... Spotify Web Player Wau The Most Amaz... Coursera P2P

PLM Recursion.logo / Tree Se connecter Java Français

World Objective



Éditeur de code: Réinitialiser API

```
public void tree(int steps, double length, double angle, double shrink) {  
    // ...  
}
```

Résultats de l'exécution

Cas de tests:

tree(10,100,90,0.68)	▶	tree(10,80,45,0.7)	▶
tree(7,75,15,0.8)	▶	tree(7,75,30,0.8)	▶

# Presentation of PLM

## Supported languages

- English
- French
- Brazilian Portuguese

# Presentation of PLM

Supported programming languages



# Presentation of PLM

## Technologies used

- Formerly a fat client
  - Written in Java
- Now a web application
  - Server implemented in Scala using PlayFramework
  - User interface written in Javascript using AngularJS and Foundation

# Presentation of PLM

Application's architecture

## 1 Presentation of PLM

- Purposes
- Demo
- About PLM
- Architecture

## 2 User's code's assessment

- Challenges
- Extraction of the execution component
- Isolation and virtualization

## 3 Result

## 4 Next steps



# User's code's assessment

## Execution components

# User's code's assessment

## Limits

- How to protect ourselves from users' rookie mistakes?
  - Infinite loops

# User's code's assessment

## Limits

- How to protect ourselves from users' rookie mistakes?
  - Infinite loops
- And from more malicious "mistakes"?
  - Infinite thread creation
  - Storage jamming with files

# User's code's assessment

## Limits

- How to protect ourselves from users' rookie mistakes?
  - Infinite loops
- And from more malicious "mistakes"?
  - Infinite thread creation
  - Storage jamming with files
- And from *System.exit(whatever)*?

# User's code's assessment

## Limits

- How to protect ourselves from users' rookie mistakes?
  - Infinite loops
- And from more malicious "mistakes"?
  - Infinite thread creation
  - Storage jamming with files
- And from *System.exit(whatever)*?
- Need also to scale.

# User's code's assessment

Judges



## 1 Presentation of PLM

- Purposes
- Demo
- About PLM
- Architecture

## 2 User's code's assessment

- Challenges
- Extraction of the execution component
- Isolation and virtualization

## 3 Result

## 4 Next steps

# Result

Live-session in TELECOM Nancy

- 30 hours of live testing with 100 students.



# Result

Live-session in TELECOM Nancy

- 30 hours of live testing with 100 students.
- Engine is (almost) working fine...

# Result

Live-session in TELECOM Nancy

- 30 hours of live testing with 100 students.
- Engine is (almost) working fine...
- ... but user experience needs to be improved!

# Result

Live-session in TELECOM Nancy

- Can't cope with the workload.

# Result

Live-session in TELECOM Nancy

- Can't cope with the workload.
- No tools for monitoring set up...

# Result

Live-session in TELECOM Nancy

- Can't cope with the workload.
- No tools for monitoring set up...
- ... so the bottleneck is unknown.

## 1 Presentation of PLM

- Purposes
- Demo
- About PLM
- Architecture

## 2 User's code's assessment

- Challenges
- Extraction of the execution component
- Isolation and virtualization

## 3 Result

## 4 Next steps

# Next steps

## Refactor the code

- Rushed to release a stable version before September...
- Needed to refactor some parts of the code.
- Standardized behavior of local and server mode.

# Next steps

Simplify workflow to adapt the content

- Store most content inside PLM.
- Heavy and error prone workflow.
- Need to extract the content from PLM's jar.
- Allow to implement an exercise editor.



# Next steps

Solve performance issues

- Set up some monitoring tools.
- Perform some load testing to identify the bottleneck.

Thanks for your attention, any questions?